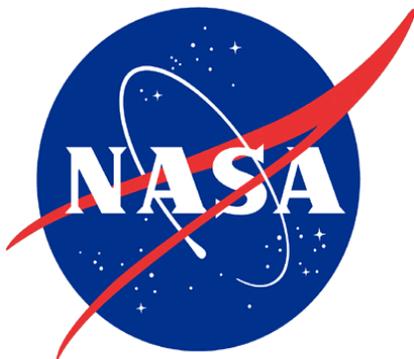

Workbook 5: Using the Huntsville Operations Support Center (HOSC) Computation Generation and Operation Applications

HOSC Training Division

October 1999



National Aeronautics and
Space Administration

George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

Workbook 5:

**Using the Huntsville Operations Support Center
(HOSC) Computation Generation and Operation
Applications**

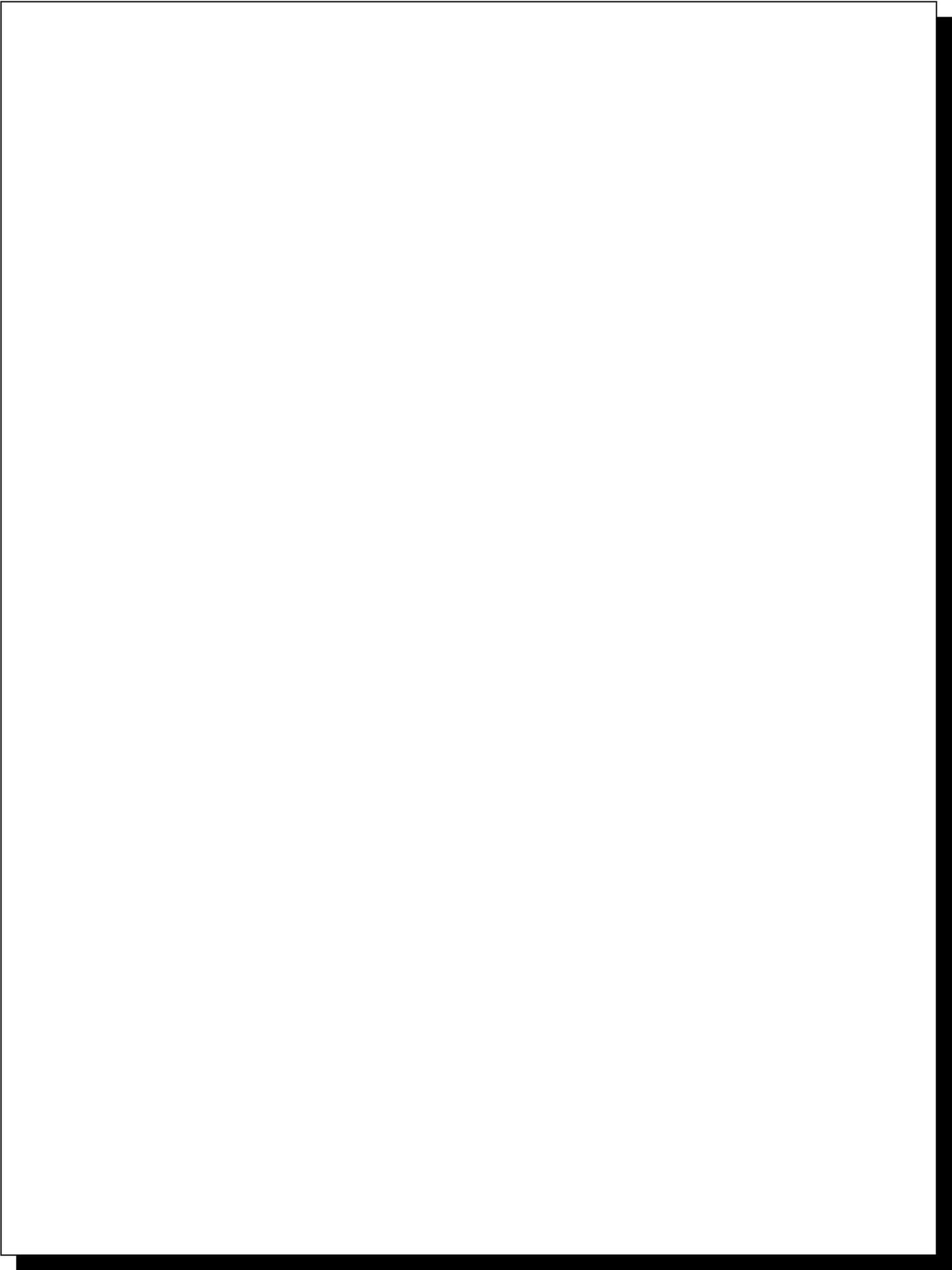
HOSC-WKBK-2080

October 1999

Prepared in support of:
NAS8-44000

Prepared by:
HOSC Training Team

Prepared for:
Mission Systems Division
Mission Operations Laboratory
Science and Engineering Directorate
Marshall Space Flight Center





Welcome

Welcome to Huntsville Operations Support Center (HOSC) training provided by the HOSC Training Team (HTT).

If you are interested in scheduling additional training, submit a training request form via the Internet. The homepage can be accessed through a secure connection at:

<https://red-dwarf.msfc.nasa.gov/webdoc/training/htt.html>

Another option is to contact the HOSC Training Coordinator, Cindy Jorgensen, at (256) 461-4927.

Training Contacts

If you have questions regarding training, contact your EO62 training representative or one of the following individuals:

Name	Organization	Phone Number	E-mail Address
Lisa Watson	EO39	544-3523	lisa.watson@msfc.nasa.gov
D.C. (Sparky) Goodman	NTI	461-4545	sparky.goodman@msfc.nasa.gov
Michelle Adrian	NTI	461-4514	michelle.adrian@msfc.nasa.gov
Gene Sobczynski	NTI	461-4516	eugene.sobczynski@msfc.nasa.gov
James (Jay) Washburn	NTI	461-4543	james.washburn@msfc.nasa.gov
Cindy Jorgensen	NTI	461-4927	cindy.jorgensen@msfc.nasa.gov

Workbook Overview

The majority of the workbooks are designed to be self-paced requiring very little assistance from an instructor. The following table lists the workbooks and their associated course number:

Workbook Numbering System

Workbook Number	Title	Associated Course
Workbook 1	HOSC End-User Software Training	HOSC-1000 HOSC-1010 HOSC-1020 HOSC-1030
Workbook 2	Using the Databases	HOSC-2050
Workbook 3	Using the Exception Monitor Application	HOSC-2060
Workbook 4	Using the Display Generation and Operation Applications	HOSC-2070
Workbook 5	Using the Computation Generation and Operation Applications	HOSC-2080
Workbook 6	Using the Scripting Applications	HOSC-2090
Workbook 7	Using the NRT Data Request Applications	HOSC-2100

Workbook Numbering System (Continued)

Workbook Number	Title	Associated Course
Workbook 8	Using Applix	HOSC-2110
Workbook 9	Using FrameMaker	HOSC-2130
Workbook 10	Using Electronic Mail	HOSC-2140
Workbook 11	Using the Payload Information Management System (PIMS)	HOSC-2150 HOSC-2160
Workbook 12	Using the Pixmap Editor	HOSC-2170
Workbook 13	Using the Ground Support Equipment Packets Application	HOSC-2180
Workbook 15	Using the End-User Command Applications	HOSC-2120
Workbook 22	Workstation Overview and General Purpose Utilities Training	HOSC-1040
Privileged Applications		
Workbook 16	Using the Privileges within the Database Applications	HOSC-3000
Workbook 17	Using the Command System Management Application	HOSC-3010
Workbook 18	Using the User Configuration Management Application	HOSC-3020
Workbook 19	Using the System Monitor and Control Applications	HOSC-3030 HOSC-3040
Workbook 20	Using the Data Packet Generator Application	HOSC-3050
Workbook 21	Using the Database Monitor and Control Application	HOSC-3060

This workbook covers course:

HOSC-2080 - Using the Computation Generation and Operation Applications

This course will provide the trainee with the information necessary to develop customized computations using the Computation Generation application and execute computations to manipulate data using the Computation Operation application.

Workbook Layout

The workbooks include a welcome section that details how the document is divided into modules as well as what is contained within each module. The modules include a discussion of the main topic of the module, a step-by-step “Try It...” and review questions. Modules have been included that “put it all together” and provide exercises to reinforce what you have learned.

This particular software is provided to assist you, the user, in the development and execution of computations which can manipulate telemetry values for use with other HOSC applications. This course will provide a thorough introduction to the purpose and structure of the software and will set you on a path toward developing your own computations.

It is required that you have some experience programming in C prior to taking this course.

The workbook and the accompanying course are divided into eight modules. At the end of each module, you are presented with a small number of review questions which allow you to test yourself on the important points covered in that module. In addition, several modules also contain scenarios followed by step-by-step “Try It...” exercises which will give you hands-on experience using the **Computation Generation/Operation** software.

Each module features an exercise which will apply the skills you acquire during the course. The eight modules include:

- Module 1: *An Introduction to Computations*
- Module 2: *A First Look at the Computation Generation Application*
- Module 3: *Building Your First Computation*
- Module 4: *More About the Computation Operation Application*
- Module 5: *Creating a Real-Time Computation*
- Module 6: *Seeing the Results of Your Computation*
- Module 7: *Getting a NRT List File*
- Module 8: *Running a NRT Computation*

Given the extensive technical requirements, extreme effort has been taken to make the software as intuitive and user-friendly as possible. Hopefully, your experience with the software will be rewarding, and you’ll find it satisfies your individual needs.

Contents

Section	Page
Welcome	i
Training Contacts	ii
Workbook Overview.....	ii
Workbook Layout.....	iv
Workbook Objectives	ix
Module 1	
An Introduction to Computations	1-1
System Initialization	1-2
The Launchpad	1-2
Message Handler.....	1-3
Starting Computation Generation.....	1-4
Starting Computation Operation.....	1-5
Module 2	
A First Look at the Computation	
Generation Application	2-1
Creating an Input MSID List	2-2
Creating an Output MSID List	2-3
Developing Your Source Code.....	2-4
Building the Executable	2-5
Module 3	
Building Your First Computation	3-1
Module 4	
More About the Computation	
Operation Application	4-1
Starting Your Computation.....	4-1
Module 5	
Creating a Real-Time Computation	5-1
Module 6	
Seeing the Results of Your Computation	6-1

Section	Page
Module 7	
Getting a NRT List File	7-1
Request Your NRT File	7-1
Module 8	
Running a NRT Computation	8-1
Look at Your NRT File	8-1
What Does it All Mean?	8-2
Starting Your NRT Computation	8-4
Course Summary	8-7
Appendix A	
Computation Generation/Operation	
Questions and Answers	A-1
Abbreviations and Acronym List	B-1
Glossary	C-1

List of Figures

Figure	Page
Figure 1-1, Acknowledge Message dialog box	1-3
Figure 1-2, Message Handler main window.....	1-4
Figure 1-3, Launchpad Generation menu	1-5
Figure 1-4, Launchpad Operation menu	1-5
Figure 2-1, Computation Generation Main Window	2-1
Figure 2-2, Input MSID List.....	2-2
Figure 2-3, Output MSID Editor	2-3
Figure 2-4, Source Editor Main Window	2-4
Figure 2-5, Build Computation Main Window	2-5
Figure 4-1, Computation Operation Main Window	4-1
Figure 4-2, Start Dialog Box	4-2
Figure 8-1, Import Dialog box	8-1
Figure 8-2, Example NRT Output in Applix	8-3

Figure

Page

Workbook Objectives

At the completion of this workbook, you should be able to:

- Identify the components of a computation
- Build simple computations
- Operate computations in different data modes

Module 1

An Introduction to Computations

Manipulating telemetry data values can be accomplished using a number of methods, including the creation and execution of computations. The HOSC provides the Computation Generation and Computation Operation applications to users to assist in building a computation. But what is a computation, and what is its purpose?

A computation is a C or FORTRAN executable created by users that can receive telemetry input data (either raw and/or pseudo data), perform operations on the input data, and then output the processed data (called pseudo data). This pseudo data can then be used by other applications internally on the same workstation or externally on different workstations.

There are two types of computations:

- User computation
- Mission computation

User computations are used to create pseudo Measurement/Stimulus Identifiers (MSIDs) which can be used on the same workstation or sent to other workstations on the project network. Users control the activation and deactivation of user computations.

Mission computations are used to create pseudo MSIDs (usually on the project server) which are then multicasted across the network to all project workstations. Most of the users for the project will be able to use the mission pseudos as needed. Only a user with a privileged account will be able to activate and deactivate mission computations. Although this type of computation is mission-specific, the process of creating such a computation is identical to that of creating the user computation.

A computation is generally made up of four components, three of which are required for a computation to function usefully:

- **Input MSID List** (required)
- **Output MSID List** (required)
- **Main Source Code** (required)
- **Subroutines and Header Files** (optional)

Using HOSC software, you can create and edit a list of input MSIDs upon which the source code will perform operations. The list of output MSIDs (pseudo MSIDs) you create are those which will carry the values produced by the operations performed upon the input MSIDs. By creating source code (including optional header files and subroutines) and specifying an Input MSID file and an Output MSID file, you can build an executable of your computation. A second HOSC application may then be used to start the computation executable in either a real-time data flow environment or using a Near Real-Time (NRT) file located on your workstation.

The next module of this workbook will introduce you to Computation Generation, a HOSC software application which allows you to create and assemble the components of a computation and build an executable.

System Initialization

When you obtain the necessary Huntsville Operations Support Center (HOSC) account, you'll be provided a username and a temporary password. These two items are required before you can login and begin using the HOSC software. When you're presented the login screen, click with the left mouse button in the **Login:** text field, type your username and press **Enter**. Next, type your temporary password and press **Enter** again. This should initialize the system and invoke the **Launchpad** and **Message Handler** applications.

Note: If the system responds with: `Login Unsuccessful`, then repeat the login procedure, making sure that you type your username and password exactly as provided. These fields are case sensitive. If you still can't gain access to the system, contact your system administrator. To change your password go to the **Utilities** menu off the **Launchpad** and click on **Change Password**.

Once you gain access to the system, you'll be presented two applications, **Launchpad** and **Message Handler**. As you'll see, all HOSC applications are designed to work together as an integrated suite of products. The number of applications you can invoke at one time is limited only by your system's memory.

The Launchpad

The **Launchpad** is the umbrella application from which all other HOSC applications are invoked. Essentially, it is a main application window. Just like other windows, it can be iconified. The **Launchpad**, however, has nothing displayed in its work area. It is initiated at system startup and cannot be closed without exiting the HOSC application software. Included in the title bar of the **Launchpad** is the name of the platform you are using, the project you are supporting, the mission, and the operational mode. The suite of applications available from the **Launchpad** is organized by function under several menu bar items. The menu bar items and individual menu items (applications) available to you depend on the project you are supporting, the type of account you requested, and your user profile. This information is maintained in the Launchpad Configuration File.

For the general user, the **Launchpad** will consist of nine menus:

- **F**ile
- **G**eneration
- **O**peration
- **D**atabase

- **PIMS**
- **PPS** or **OFLS**
- **CM Tools**
- **Utilities**
- **Help**

Privileged users may have two additional menu items:

- **Management**
- **SMAC**

Message Handler

The **Message Handler** application provides you with the capability to view messages generated by applications currently running for a particular project and mission. The **Message Handler** application is started when you login and cannot be exited, although it can be resized and iconified. All messages have several components describing the message including the time, the user Identification (ID), the message priority, the message type, and the source of the message. The **Message Handler** application reads application messages from a buffer, applies the message selection criteria, and displays or stores these messages so that you can selectively view them.

In some instances you may be required to acknowledge the receipt of messages. These critical messages will be displayed within a message dialog box as shown in Figure 1-1, Acknowledge Message dialog box.

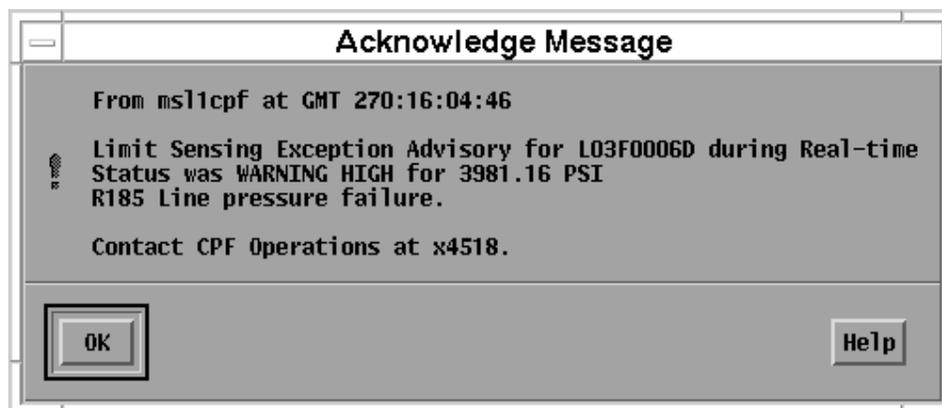


Figure 1-1, Acknowledge Message dialog box

The **Message Handler** main window allows you to view system and application messages generated by the various HOSC applications (see Figure 1-2, Message Handler main window). The **Message Handler** application is initially setup with a default message filter that is configured to display all types of messages.

Within the main window you can:

- Manipulate message filters so you see only those messages of particular interest
- Configure the display columns so that you only see those aspects of messages with which you are most concerned
- Pause or control the rate at which messages are displayed

Clear the messages currently displayed

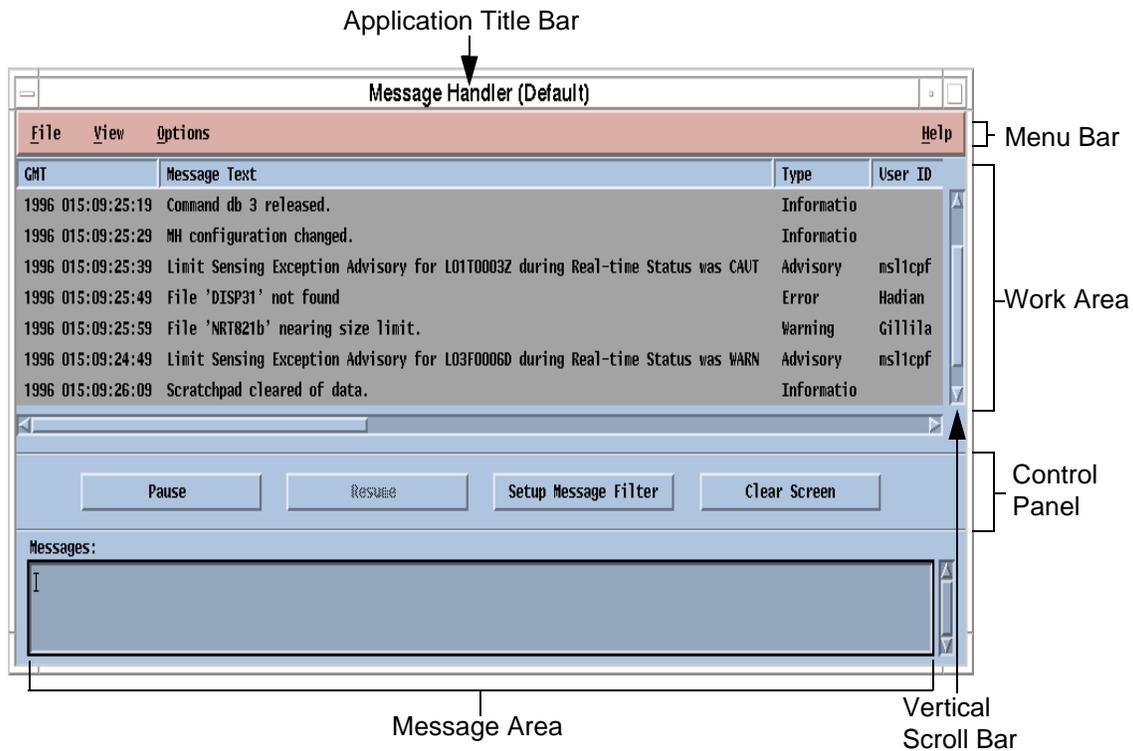


Figure 1-2, Message Handler main window

Starting Computation Generation

To invoke the Computation Generation application from the **Launchpad**, click on the **Generation** menu and then click on **Computation Generation** (see Figure 1-3, Launchpad Generation menu).

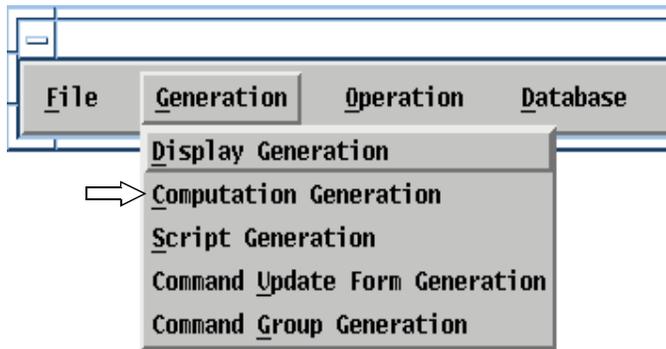


Figure 1-3, Launchpad Generation menu

Starting Computation Operation

The Display Operation application is started from the **Launchpad** by clicking on the **Operation** menu and selecting **Computation Operation** (see Figure 1-4, Launchpad Operation menu).

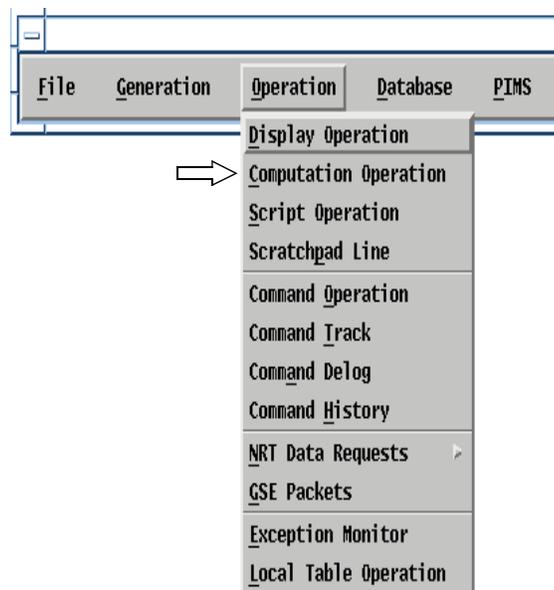


Figure 1-4, Launchpad Operation menu

Review Questions

Indicate an answer for each question below. The correct answers are given immediately following the questions.

1. What two programming languages are available for the development of computations?
2. Name and describe the two types of computations.
3. What are the four main components of a computation?
4. Of the four components, which are optional?
5. Which HOSC application is used to create, assemble and build a computation?

Review Answers

1. C and Fortran
2. User and Mission.

User computations are used to create pseudo Measurement/Stimulus Identifiers (MSIDs) which can be used on the same workstation or sent to other workstations on the project network.

Mission computations are used to create pseudo MSIDs which are then multicasted across the network to all project workstations.

3. The four components of a computation include:
 - Input MSID List
 - Output MSID List
 - Main Source Code
 - Subroutine and Header Files
4. Subroutine and header files
5. Computation generation

Module 2

A First Look at the Computation Generation Application

The Computation Generation application allows you to create and edit source code files, input Measurement/Stimulus Identifier (MSID) and output MSID files. You can then assemble and validate these components and compile them into an executable.

Once you have created an executable version of a computation, this executable can be initiated using a sister application called Computation Operation, which is covered later in the course.

Let's open the Computation Generation application on your workstation. From the Launchpad, select the **Generation** menu. From the menu, select **Computation Generation**.

The main window of the application serves the singular purpose of providing access to the four mini-applications which you use to build the components of a computation (see Figure 2-1, Computation Generation Main Window). After you initiate the application, you will see the following window appear:

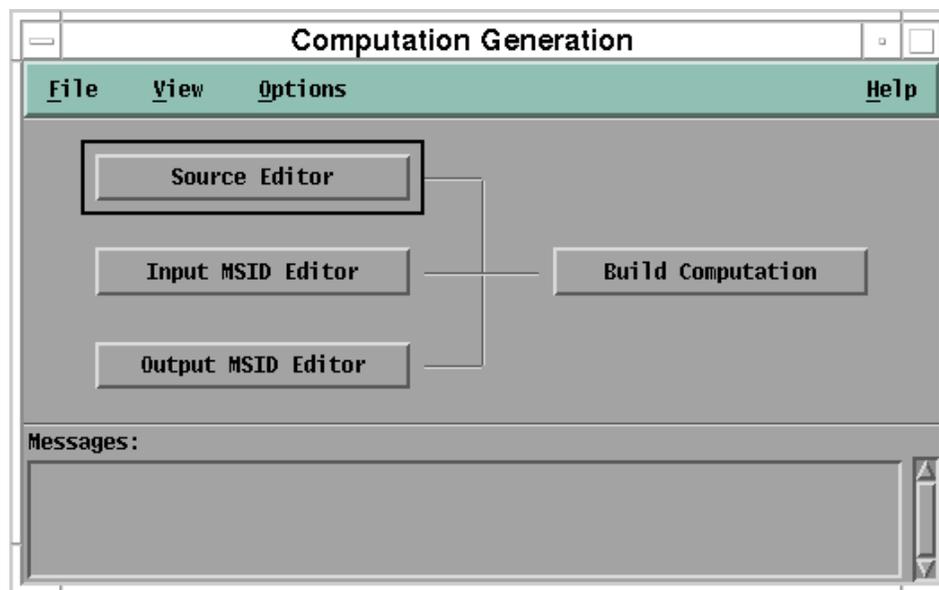


Figure 2-1, Computation Generation Main Window

Notice the four pushbuttons which allow you to invoke the four mini-applications which make up Computation Generation:



- invokes the Source Editor mini-application which allows you to create source code for your computation in either

C or FORTRAN programming languages. This can be either your main source or subroutines.



- invokes the Input MSID Editor mini-application which allows you to create a list of input MSIDs that will be used in your computation.



- invokes the Output MSID Editor mini-application which allows you to create a list of output MSIDs that will be used in your computation.



- invokes the Build Computation mini-application which allows you to assemble the components of your computation and build an executable file from your sources.

Creating an Input MSID List

The first step toward creating a computation is often deciding which parameters from the telemetry data stream you need to manipulate and monitor. You can use the Input MSID Editor mini-application (which is invoked by pressing the appropriate button on Computation Generation's main window) to search the telemetry database and create an Input MSID list. This list, when used in conjunction with your source code, output MSID list and subroutines, can be used to generate a computation executable.

Once this mini-application is invoked, the following window appears:

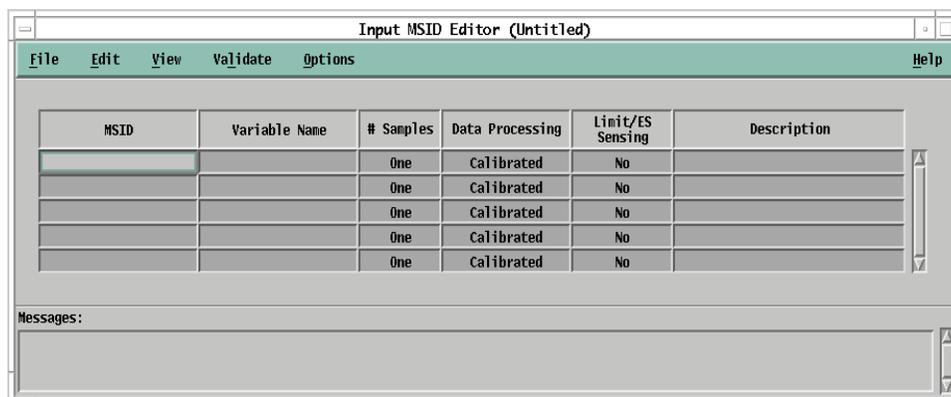


Figure 2-2, Input MSID List

Input MSIDs can be a mixture of raw (downlinked), External (\$E) or Internal (\$I) pseudo MSIDs, and Input Computation Constants (\$C). This allows you to create a computation that obtains values from telemetry sources, computation sources, and/or application sources.

External pseudo MSIDs are multicasted on the network to be accessed by all applications on the project network. Using **\$E** as the first characters of an MSID name signifies that it is an External MSID. These pseudos must be defined in the Telemetry Database (TDB).

Internal pseudo MSIDs allow local applications to share real-time pseudo data values without placing decommutation, conversion, and calibration information into the database. Using **\$I** as the first characters of an MSID name signifies that it is an Internal MSID. These internal values are not multicasted across the network as external pseudos and are therefore not available to other users.

The Input Computation Constant allows you to define global variables with default values. Using **\$C** as the first characters of an MSID name signifies that it is a Computation Constant. These values also cannot be shared with other users.

Creating an Output MSID List

Once you have created your list of input parameters, the next logical step is to create a list of the pseudo-MSIDs that will carry your processed values. You can use the Output MSID Editor mini-application to create an Output MSID list. Remember that you can only produce pseudo values (i.e., external or internal pseudo-MSIDs). Internal pseudos can be of your own creation; however, external pseudos must be available in the telemetry database.

Once this mini-application is invoked, the following window appears:

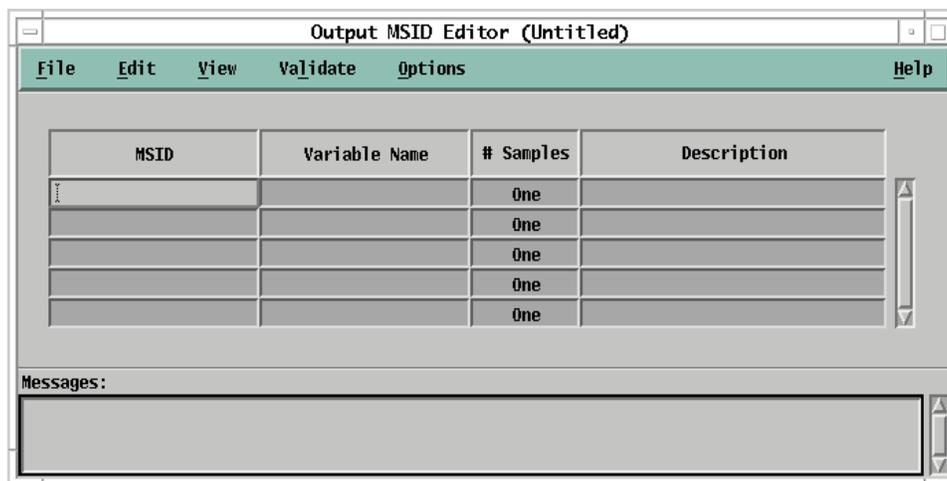


Figure 2-3, Output MSID Editor

Output MSIDs can be either External (**\$E**) or Internal (**\$I**) pseudo MSIDs since only pseudo MSIDs can be produced as output from a computation.

Developing Your Source Code

After deciding which values to manipulate and creating a list of output values to carry your processed information, you should be ready to create the source code which will perform the actual processing. Whether you prefer C or FORTRAN, you can develop source code using the Source Code Editor mini-application.

The HOSC software makes source code development as simple as possible by providing templates which guide you through the process. Also provided is a pre-compiler that supplies source code which is needed to support data retrieval and decommutation, pseudo MSID initialization, and other pre-compilation functions.

Invoking the application causes the Source Editor's main window to appear on your screen:

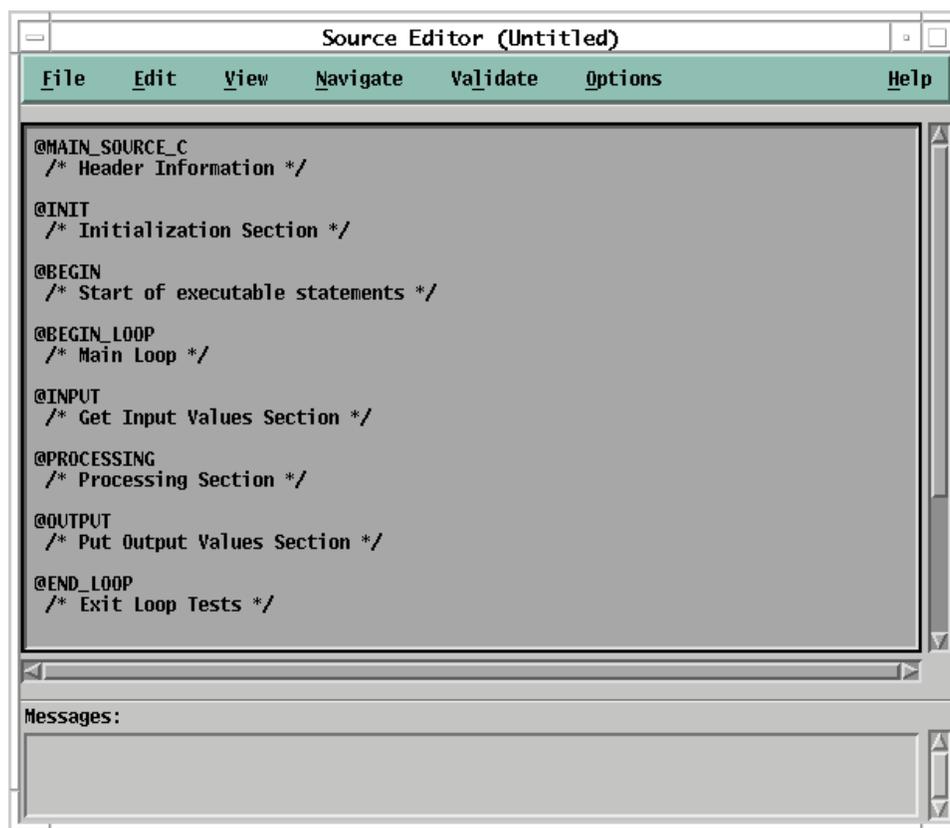


Figure 2-4, Source Editor Main Window

You can create a number of different file types with this editor for both programming languages. To simplify the creation of these files, it is recommended that you use one of the supplied templates. File types you can create include main C code, main FORTRAN code, subroutine C code, subroutine FORTRAN code, and header files. These templates can be selected from a dialog box invoked from the **F**ile menu's **N**ew... menu item.

Building the Executable

Finally, when all of the computation components are created, you can use the Build Computation mini-application to assemble and validate the components and create an executable version of the computation. This mini-application allows you to generate an executable file by selecting a source code file, an input MSID file, an output MSID file, and any needed subroutines. This capability to create independent files and then link them together for building the computation allows you to use Input/Output MSID files with multiple computations.

Invoking this mini-application displays the following window:

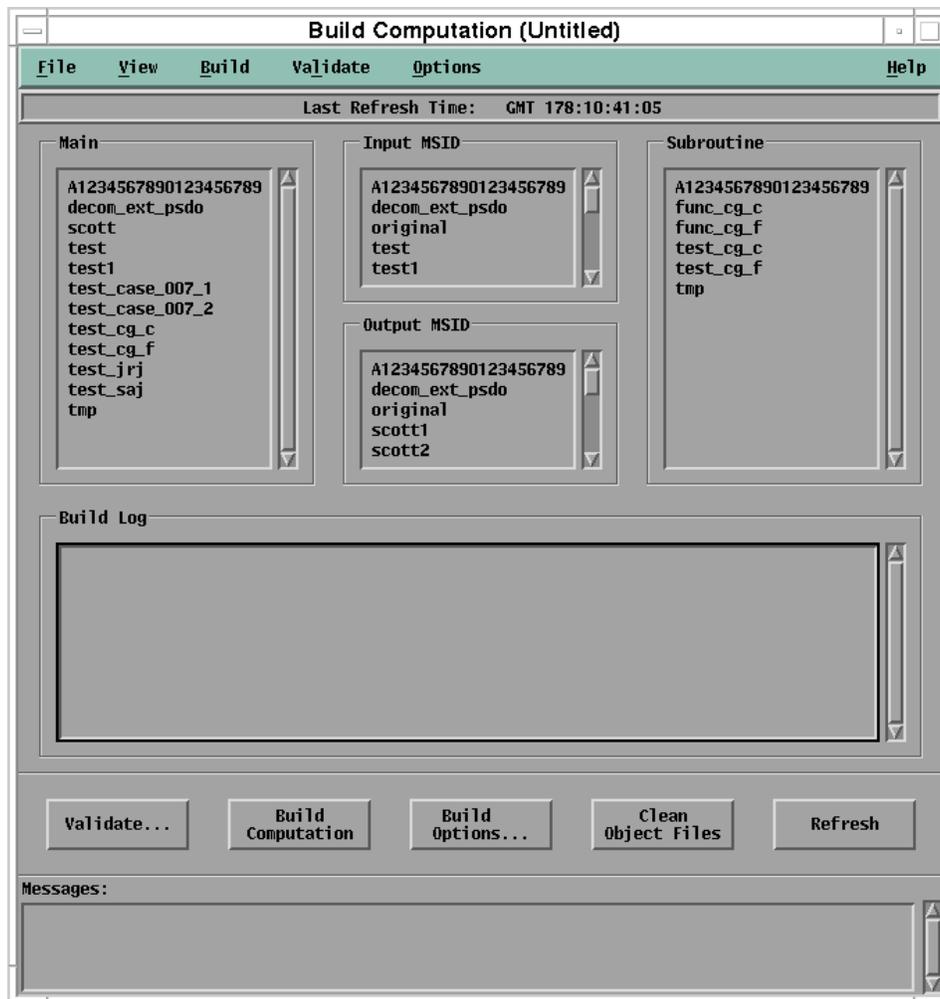


Figure 2-5, Build Computation Main Window

Select the components of your computation and validate them. You can then set any build options (such as linked libraries and code optimization levels) before building the actual executable. Once the executable is created, you can use the Computation Operation application to initiate the computation itself.

Review Questions

Indicate an answer for each question below. The correct answers are given immediately following the questions.

1. Name the four mini-applications which make up the Computation Generation application.
2. What is the purpose of an Input MSID list?
3. What is the purpose of an Output MSID list?
4. What file types can be created using the Source Code editor?
5. Which mini-application allows you to assemble, validate and compile your computation so that it can be executed?

Review Answers

1. Input MSID editor, Output MSID editor, Source editor, and Build computation
2. To define those MSIDs from the telemetry data stream you need to manipulate and monitor.
3. To define the pseudo MSIDs that will carry the processed values.
4. Source C and Fortran.
Main source subroutine.
Header files.
5. Build computation

Module 3

Building Your First Computation

Now that you are familiar with the Computation Generation application, it is a good idea to apply this knowledge to the process of actually creating a working computation.

Although the computation you will create in this module of the course is simple, going through this module will familiarize you with the creative process that applies to building all computations.

Exercises

Instructions

The following “Try It” allows you to develop a simple computation. Carefully read and complete each step.

Scenario

You are responsible for monitoring an experiment onboard the space vehicle that must operate under specific temperature and pressure conditions. Your display features a warning pixmap that should change any time either the temperature or pressure of your experiment environment exceeds certain warning limits. This would be easy enough to represent if only one parameter needed to be monitored using the one pixmap object. However, since two parameters must be checked, you need to write a computation that checks both the temperature and pressure and then makes changes to the pixmap based on that criteria.

Try It...

1. Click the **Generation** menu on the **Launchpad** and select **Computation Generation**.
2. From the **Computation Generation** main window, click **Input MSID Editor**.
3. Now refer to the project worksheet to for the list of MSIDs to be used. In the first row, under the **MSID** column heading, enter the first input MSID. Under the **Variable Name** heading, enter “temp”. Under the **Description** heading, enter “Temperature”.

Remember that variable names should not contain trailing spaces and that MSIDs are case-sensitive and should not contain trailing spaces.

4. In the second row, under the **MSID** column heading, enter the second input MSID (see handout). Under the **Variable Name** heading, enter “pressure”. Under the **Description** heading, enter “Pressure”.

5. From the **File** menu on the **Input MSID Editor**, click **Save**. In the dialog box, use the filename “test2msid_XXX”, where XXX are your initials. Click **Save**.
6. Close the **Input MSID Editor**.
7. On the **Computation Generation** main window, click **Output MSID Editor**.
8. In the first row, under the **MSID** column heading, enter the output MSID from the project worksheet. Under the **Variable Name** heading, enter “outcode”. Under the **Description** heading, enter “Test Results”.

Remember that MSIDs and variable names are case-sensitive and should not contain trailing spaces.

9. From the **File** menu on the **Output MSID Editor**, click **Save**. In the dialog box, use the filename “test2msid_XXX”, where XXX are your initials. Click **Save**.
10. Close the **Output MSID Editor**.
11. On the **Computation Generation** main window, click **Source Editor**.
12. From the **File** menu on **Source Editor**, select the **New...** menu item. In the dialog box that appears, select “Main_C” as your template and click . The work area of the Source Editor window will be replaced by a template. Enter the following source code so that it matches the on-screen source code. You will not need to enter any of the lines that begin with the @ symbol. You are also not required to enter any of the comment lines (which begin with /*). However, for the sake of keeping your source code clear to yourself and other readers, commenting is strongly recommended.

Remember while entering source code that whitespace is ignored in C code except when it appears between quotation marks. Also, remember that C keywords (such as “printf”) are case-sensitive.

```
@MAIN_SOURCE_C
```

```
/* Header information */
```

```
/* This is a computation which tests two */  
/* parameters' warning limits and sets */  
/* an external MSID flag if warning limits */  
/* are exceeded */
```

```

@INIT
/* Initialization section */

int stat, get_status, outcode;
short temp;
unsigned short pressure;

@BEGIN
/* Start of executable statements */

@BEGIN_LOOP
/* Main Loop */

@INPUT
/* Get Input Values Section */

get_status = CPS_Get_Next_Value(temp);
get_status = CPS_Get_Next_Value(pressure);

@PROCESSING
/* Processing Section */

outcode = 0 ; /* Resets output value to nominal */

if ((temp<-32 || temp>98) || (pressure<45 ||
    pressure>77))

    outcode = 1 ;

stat = CPS_Put_Pseudo_Value(outcode, 1);

@OUTPUT
/* Put Output Values Section */

printf("Pseudo Output: %d\n",outcode);

@end_LOOP
/* End Main Loop */

@END
/* Termination */

@FUNCTIONS

/* Subroutines */

```

Note that there may be additional sections beginning with the “@” symbol on your screen that are not included in the computation source listed above. As the software progresses from build to build, new sections may be added. Leave these sections in your computation if they appear on your screen and speak to your instructor if you have any questions about these additional sections.

13. Select the **File** menu from the **Source Editor** main window and click **Save As...** In the dialog box, use the filename “test2msid_xxx”, where xxx are your initials. Click **Save**.
14. Close the **Source Editor**.
15. On the **Computation Generation** main window, click **Build Computation**.
16. In each frame of the dialog box where file names are listed, click on the file which you created in the previous steps. You must select the input MSID file, the output MSID file, and the source code file. Click the **Validate...** menu item.
17. Enter a filename (e.g., “test2msid_xxx”, where xxx are your initials) for the executable and description file in the **Save As** dialog box. Click **Save**.
18. Click **Validate...** again to invoke the **Validate** dialog box.
19. Click **Validate**. Once validation is complete and error-free, close the dialog box.
20. Click **Build Computation**. Once build is complete, close the dialog box.

Execute It...

You will use the Computation Operation application to execute the computation you just created. Although this application will be covered in detail in the next module, following these steps will allow you to view the output of your computation. You may not understand some of the concepts at this point, but you will at least see how a computation executes.

1. Click the **Operation** menu on the **Launchpad** and select **Computation Operation**.
2. Click .
3. In the **Computations** frame of the **Start** dialog box, select the computation you created earlier in **Computation Generation**. If you

followed the recommended filename structure, you should see a file called “test2msid_xxx” where xxx are your initials.

4. In the **Input/Output Type** frame, select the **Packet** radio button.
5. In the **Data Mode** frame, select the **Real-time** radio button.
6. In the **Database Version** frame, select the latest revision of the TDB.
7. In the **Priority** frame, select the **Medium** radio button.
8. Click .

If all goes as planned, your computation should begin executing and you will see text output (including the value of the pseudo-MSID you are writing to) in the Standard Output area of the Computation Operation main window.

When you are ready to stop the computation, click on the name of the computation in the work area of Computation Operation. This causes  to become available at the bottom of the window. Click this button to stop the computation.

The next module of this workbook will give you a better look at the Computation Operation application. You will also be introduced to the various data modes under which a computation can be run.

Review Questions

Indicate an answer for each question below. The correct answers are given immediately following the questions.

Most of these question are theory based and do not necessarily cover material from the workbook. Most of these concepts are covered in the user guide.

1. What is the most logical order for building the components of a computation? Is it a requirement to build computation components in that order?
2. Are MSIDs case-sensitive? What about variables in your code?
3. Are templates required for creating computations? Are they recommended? Why or why not?
4. When you first attempt to validate your computation within the Build Computation mini-application, a **Save As** dialog box appears. What two files are you specifying by this filename? What is the purpose of each?
5. Why is it possible to apparently give the same file name to each component of a computation? Why is it often a good idea to do so?

Review Answers

1. Inputs, outputs, source code. You do not have to develop the components in any certain order.
2. Case-sensitivity as applied to MSIDs varies from application to application; therefore it would be prudent and far less confusing to consider MSIDs as being case-sensitive. Variables are case-sensitive.
3. No they are not required for creating computations. They are recommended since they minimize coding time.
4. The description file and the executable. The description file contains the build options used and the files included. The executable is used by Computation Operation to initiate the computation.
5. Because each file is saved in a separate sub-directory. So that it is easy to identify the components of a specific computation.

Module 4

More About the Computation Operation Application

In the previous module, you were quickly guided through the simple process of starting a computation using the Computation Operation application. This module will provide a full review of this particular application's capabilities as well as introduce the various data modes and input environments in which your computations can run.

Starting Your Computation

Once you've created an executable version of your computation, you can use the Computation Operation application to start the computation on your workstation. The main window appears as follows:

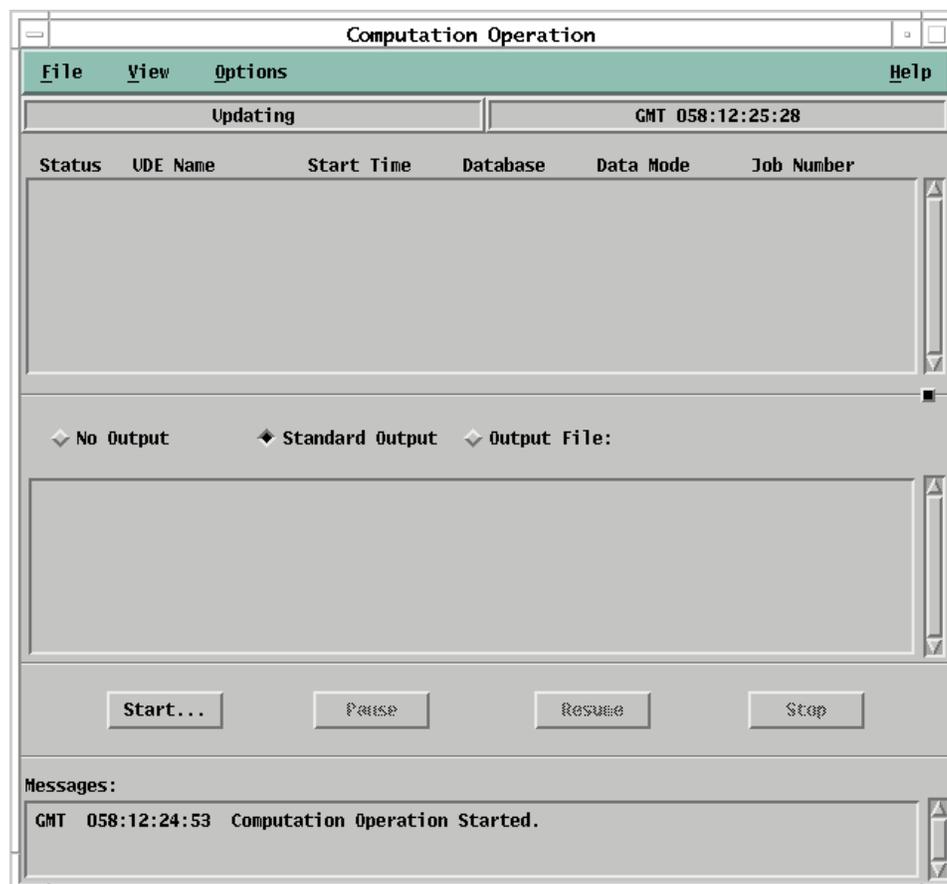


Figure 4-1, Computation Operation Main Window

Starting a computation is as simple as clicking and selecting a few options, including the input/output type, the database version and the priority of CPU time given to the computation.

Input/output types encompass two main modes of operation. In a real-time environment when telemetry data is available on your workstation, you can select from real-time, dump, and playback data packets. However, if you need to test a computation or use a computation on previously recorded data, you can request a NRT file and run the computation using that file as input. You can also elect to send the output of a computation to a file.

The input/output type (and its related options) as well as the CPU priority and database version is selected from the following dialog box which is invoked by clicking on the main window:

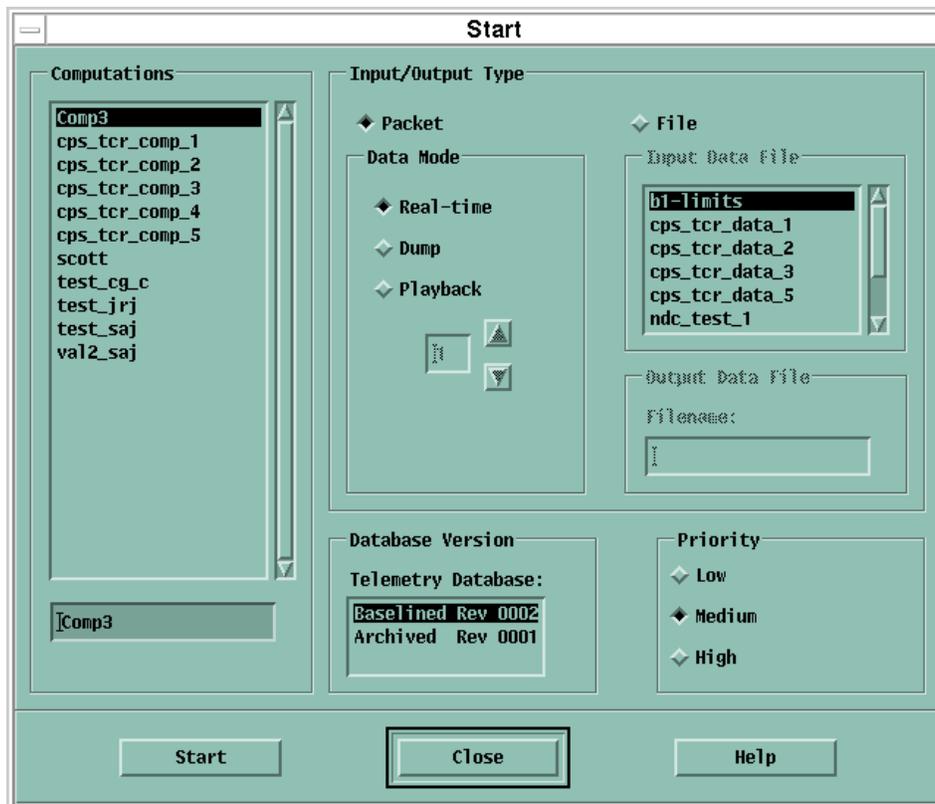


Figure 4-2, Start Dialog Box

Clicking on this dialog box actually initiates the computation. You can then monitor, pause/resume, and stop the computation from the main window.

Seeing the computation's output in the Standard Output area is sufficient enough to see that the computation is running correctly for a single-MSID output computation like you created in the previous module. However, the next two modules allow you to create a useful (and more complex) computation and then create a display which provides graphic monitoring of the computation's output.

Review Questions

Indicate an answer for each question below. The correct answers are given immediately following the questions.

1. Which button on Computation Operation's main window do you click to start a computation?
2. What two input environments are available?
3. In a situation where telemetry data is available, what modes of data are you able to select from?
4. What kind of file is needed to run a computation from an input file?
5. What is the purpose of the **Priority** frame on the **Start** dialog box?

Review Answers

1. The Start pushbutton.
2. Packet or file
3. Real-time, dump, playback 1-13.
4. The output from a NRT List Request.
5. The Priority allows you to determine the processing priority of a computation in relation to all other active computations. The valid options are:

Low - Provides the least amount of processing time to the computation in regard to other computations that are simultaneously active

Medium - Provides the average amount of processing time to the computation. This is the default setting.

High - Provides the most amount of processing time to the computation in regard to other computations that are simultaneously active.

Module 5

Creating a Real-Time Computation

The last three modules have prepared you for the creation of a useful and more complex computation. This module will present you with such a task. You will find, however, that the process for creating a computation, no matter how complex, will be almost identical from one computation to the next.

This computation will be run against real-time data. In other words, the purpose of this computation is to read input from the telemetry data stream, process a number of values, and output these values as external pseudos. In modules 7 and 8 you will use this computation again in coordination with an input file in near real-time mode (NRT).

Exercises

Instructions

The following “Try It” allows you to develop a simple computation. Carefully read and complete each step.

Scenario

Certain parameters in the data stream carry values in a format that can only be used with certain objects on a display. You need to write a computation that reads in these parameters and converts these values into pseudo-values in a format usable by other display objects.

Try It...

C programmers will notice in the source code below that part of this process is accomplished by using the **union** keyword.

1. Click the **Generation** menu on the **Launchpad** and select **Computation Generation**.
2. From the **Computation Generation** main window, click **Input MSID Editor**.
3. Now refer to the MSID project worksheet. Fill out the on-screen Input MSID form according to the corresponding table on the handout.

Remember that MSIDs are case-sensitive and should not contain trailing spaces. Also, variable names should not contain trailing spaces.

4. From the **File** menu on the **Input MSID Editor**, click **Save**. In the dialog box, use the filename “mod5_XXX”, where XXX are your initials. Click **Save**.
5. Close the **Input MSID Editor**.

6. On the **Computation Generation** main window, click **Output MSID Editor**.
7. Referring again to your handout, fill out the Output MSID form using the corresponding table.
8. From the **File** menu on the **Output MSID Editor**, click **Save**. In the dialog box, use the filename “mod5_XXX”, where XXX are your initials. Click **Save**.
9. Close the **Output MSID Editor**.
10. On the **Computation Generation** main window, click **Source Editor**.
11. From the **File** menu on **Source Editor**, select the **New...** menu item. In the dialog box that appears, select “Main_C” as your template. The work area of the Source Editor window will be replaced by a template. Enter the following source code so that it matches the on-screen source code. You will not need to enter any of the lines that begin with the @ symbol. You are also not required to enter any of the comment lines (which begin with /*). However, for the sake of keeping your source code clear to yourself and other readers, commenting is strongly recommended.

```
@MAIN_SOURCE_C
```

```
/* Header information */
```

```
/* This is a computation to convert */  
/* input values to a different format */  
/* for a display */
```

```
@INIT
```

```
/* Initialization section */
```

```
int comp_status, aout;  
short ain, bin;  
unsigned short cin, din, ein, fin;  
char bout, cout, dout, eout, fout;  
int ain_get_status, bin_get_status, cin_get_status, din_get_status = 0;
```

```

int ein_get_status, fin_get_status = 0;

union u_type {
    int i;
    short s;
};
union u_type a,b,c,d;

@BEGIN

/* Start of executable statements */

@BEGIN_LOOP
/* Main Loop */

@INPUT

/* Get Input Values Section */

ain_get_status = CPS_Get_Next_Value(ain);
bin_get_status = CPS_Get_Next_Value(bin);
cin_get_status = CPS_Get_Next_Value(cin);
din_get_status = CPS_Get_Next_Value(din);
ein_get_status = CPS_Get_Next_Value(ein);
fin_get_status = CPS_Get_Next_Value(fin);

@PROCESSING

/* Processing Section */

a.i = ain;
a.out = a.s;
b.i = bin;
b.out = b.s;
c.i = cin;
c.out = c.s;
d.i = din;
d.out = d.s;
e.out = ein;
f.out = fin;

comp_status = CPS_Put_Pseudo_Value(a.out, 1);
comp_status = CPS_Put_Pseudo_Value(b.out, 1);

```

```

comp_status = CPS_Put_Pseudo_Value(cout, 1);
comp_status = CPS_Put_Pseudo_Value(dout, 1);

@OUTPUT

/* Put Output Values Section */

printf("Pseudo Output(2*int): aout:% 12i bout:% 12i\n", aout, bout);
printf("Pseudo Output(2*int): cout:% 12i dout:% 12i\n", cout, dout);
printf("Pseudo Output(2*flt): eout:% 12g fout:% 12\n", eout, fout);

@END_LOOP

/* Exit Loop Tests*/
/* End Main Loop */

@END

/* Termination*/

@FUNCTIONS

/* Subroutines */

```

12. Select the **File** menu from the **Source Editor** main window and click **Save As...** In the dialog box, use the filename "mod5_xxx", where xxx are your initials. Click **Save**.
13. Close the **Source Editor**.
14. On the **Computation Generation** main window, click **Build Computation**.
15. In each frame of the dialog box where file names are listed, click on the file which you created in the previous steps. Click the **Validate...** menu item.
16. Enter a filename (e.g., "mod5_xxx", where xxx are your initials) for the executable and description file in the **Save As** dialog box. Click **Save**.
17. Click **Validate...** again to invoke the **Validate** dialog box.
18. Click **Validate**. Once validation is complete and error-free, close the dialog box.

19. Click **Build Computation**. Once build is complete, close the dialog box.

Execute It...

Now you will use the Computation Operation application to execute the computation you just created.

1. Click the **Operation** menu on the **Launchpad** and select **Computation Operation**.
2. Click .
3. In the **Computations** frame of the **Start** dialog box, select the computation you just created. If you followed the recommended filename structure, you should see a file called “mod5_xxx” where xxx are your initials.
4. In the **Input/Output Type** frame, select the **Packet** radio button. This allows you to run the computation in a real-time environment when telemetry data is available to your workstation.
5. In the **Data Mode** frame, select the **Real-time** radio button. We will use the currently flowing real-time data present in the telemetry data stream.
6. In the **Database Version** frame, select the latest revision of the TDB.
7. In the **Priority** frame, select the **Medium** radio button.
8. Click .

If all goes as planned, your computation should begin executing and you will see text output (including the value of the pseudo-MSIDs you are writing to) in the Standard Output area of the Computation Operation main window. Because you are actually writing out a number of values, some of these values may be difficult to keep up with in this window.

This difficulty in monitoring your computation’s progress leads to the next module which will guide you through creating a display which will allow you to view your computation’s output in a more convenient fashion.

Review Questions

Indicate the answers for each question below. The correct answers are given immediately following the questions.

These questions invite you to think about how to go about debugging a computation you have written when errors are reported during the validation process. Other questions involve the building and execution processes. These topics are covered in the user guide.

1. While validating your computation, the software reports that it cannot understand some of the syntax used in your source code. Which mini-application would you use to correct this problem?
2. While validating your input MSID list, the software alerts you to a parameter that cannot be found in the database. When you return to the Input MSID Editor, which menu item could be used to ensure that the MSID is correctly spelled and does exist in the selected database?
3. You need to link your C code to an external math library. On which dialog box (and in which application) can you specify this?
4. You want to make sure your computation gets the maximum amount of CPU time possible. How do you specify this, and in which application?
5. You leave the Computation Operation application but find that your computation is still running. How do you stop it?

Review Answers

1. Source Editor.
2. Select MSID.
3. Build Computation on the Build Options dialog box.
4. By changing the priority to Highest on the Build Options dialog box.
5. Restart the Computation Operation application, select the computation, and click stop. You can also stop the process by selecting Process Manager from the Utilities menu, highlight the process and select Kill from the Command menu.

Module 6

Seeing the Results of Your Computation

The Standard Output area of Computation Operation's main window is sufficient for viewing the output of very simple computations. Often, however, you will want to apply the results of a computation to a more readable application. The HOSC software provides the Display Generation and Display Operation applications for the creation and execution of graphical displays which allow you to monitor telemetry data and pseudo telemetry data. Pseudo telemetry data is that data produced by computations.

This module guides you through the creation of a very simple display which displays the constantly updated values of the output from your computation.

Exercises

Instructions

The following "Try It" allows you to develop a simple computation. Carefully read and complete each step. If you are not familiar with the Display Generation application, refer to the Display Generation Tool Palette handout provided with this workbook. This sheet will identify the few tools you will need to build this display.

Try It...

1. Click on the **Generation** menu on the **Launchpad** and then on the **Display Generation** menu item.
2. Resize the display (not the window) by clicking on the boundary handle and dragging it. Release it at the desired size.
3. Create an MSID text field on the display by clicking the MSID output object tool on the tool palette. The cursor will change to a cross-hair. Click within the work area to place the measurement.
4. With the "MSID Text (?)" item selected, click the right mouse button and select **Set Object Attributes...** from the **Display Popup Menu** to invoke the **Set MSID Text Attributes** dialog box.

For the following items, refer back to your MSID Project Worksheet. You will be referring to the Output MSID list for the Module 5 computation.

5. Within the **Object Attributes** frame, insert a label for the measurement. Use the description of the first output MSID as the label.
6. Within the **Data Attributes** frame, click on  to invoke the **Select MSID** dialog box.

7. Click on and select the first MSID on your handout from the on-screen list. Click on . The MSID will be entered into the text field on the **Set MSID Text Attributes** dialog box.
8. Set **Data Processing** to **Converted**.
9. In the **Data Representation** portion of the **Data Attributes** frame, you may select whether you want to use the database default data representation or define your own format from those available for that data type. Use the default representation for this exercise.
10. In the **Recall Text** frame, type the MSID and any other comment text you'd like to display in **Display Operation**. Click on .
11. Repeat steps 5-10 for each additional MSID in your list, creating a new MSID text field for each MSID.
12. If you are familiar with the Display Generation application, you can align the text fields, alter the color of the background, and make any additional changes you desire.
13. Open the **Define** menu and select **Fit Display to Objects**. Go to the **File** menu and select **Save**. Save your display using your first name.

Execute It...

Now you will use the Display Operation application to execute the display you just created. You should have your computation running in order for the display to function correctly.

1. Click on the **Operation** menu on the **Launchpad** and select **Display Operation**.
2. Click on the **File** menu and select **Open...** Select the display you just created (designated by your first name) from the open dialog box and click on . Are your parameters and display objects updating?

Your simple display should be providing you with an easy-to-read update of the pseudo-parameters to which your computation is writing.

Review Questions

Indicate the answers for each question below. The correct answers are given immediately following the questions.

These questions invite you to think about troubleshooting.

1. Your display doesn't seem to be updating with new values. You open the Computation Operation application and see that the main window's list of active computations is empty. What do you need to do?
2. Your computation is running in playback mode while your display is configured for real-time. What happens to the parameters in your display?

Review Answers

1. Start your computation that drives the pseudos defined on your display.
2. The display will not update because the computation is running in the playback mode and you are displaying realtime data on your display.

Module 7

Getting a NRT List File

You are facing a difficult situation. Data flow is not available to your workstation and you have a computation you need to test before the end of your shift. What can you do to rectify this problem?

Computation Operation allows you to run a computation using a near real-time (NRT) file as input. This file contains information from the telemetry data stream over a certain period of time. You can use the NRT List Request application, which is available on the HOSC Launchpad, to request a file and then use Computation Operation to run your computation using this file.

This module guides you through the process of requesting a NRT file for use with the computation you created in Module 5. In the next module, you will use Computation Operation to execute the NRT computation.

Request Your NRT File

A **NRT List Request** is used to retrieve samples for a specified list of parameters for a specified time period. It contains information such as a list of parameters, the start time and stop time that define the time period, the data mode, the database version, and an optional packet ID indicating the data source. Additionally, specific information about each parameter can be specified. For example, you can specify whether to perform limit/expected state sensing on the samples to be returned, whether to perform sample thinning, how to format the samples that are to be returned, and what, if any, filter test to apply to determine whether to return a sample.

Exercises

Instructions

The following “Try It” allows you to develop a NRT List Request. Carefully read and complete each step.

Try It...

1. Click the **Operation** menu on the **Launchpad** and select the **NRT Data Requests** cascade menu. From the cascade, select **NRT List Request**.
2. In the **Name Information** frame, there are two text entry fields. In the **Request Name:** field, enter `comptest`. Your keystrokes are automatically reflected in the **Output Filename:** field.
3. In the **Time Information** frame, set the **Time Source:** to **HOSC Receipt Time**.

4. You will need to find an available NRT segment. Select **Show NRT Directory...** from the **Options** menu. Specify a start time of 001:00:00:00. Specify a stop time of 365:00:00:00. Click . Select and write down a 4 minute segment close to the current time. If you do not select a segment close to the current time, there may be no data in existence for the parameters you are going to request. After a time slice has been selected, you can close the dialog box.
5. In the **Start Time:** entry area, enter the starting GMT (DDD:12:00:00) of the time segment you selected in the fields provided (i.e., **DDD** = current julian date in the first field, **12** in the second field, etc.).
6. In the **Stop Time:** entry area, enter the ending GMT of the time segment you selected (DDD:12:10:00) in the fields provided (i.e., **DDD** = current julian date in the first field, **12** in the second field, etc.).
7. Enter the MSID information in the **MSID List Information** frame. You can either type in the following information or use the **Select MSID...** item from the **Options** menu to select your MSIDs. Enter the MSIDs from your MSID List handout into the table. Refer to the Input MSID list from Module 5. You must enter the MSIDs into the table in the same order in which they appear in your computation's input MSID list. Otherwise, the data will not be read into the computation in the correct order and the wrong MSID values will be processed.
8. For each MSID, click in the **Data Processing** column. From the **Options** menu, click **Set Data Processing**. Select **Converted**.
9. For each MSID, click in the **Data Representation** column. From the **Options** menu, click the **Set Data Representation...** menu item. In the dialog box that appears, select the **Defaults** radio button and click .
10. For each MSID, click in the **Sample Thinning** column. Click your right mouse button to invoke a popup menu and select **Set Sample Thinning...** From the dialog box, select **One** and click .
11. In the **Limit/ES Sensing** column, for each MSID, enter **No**. This can also be accomplished by clicking in the cell, clicking **Set Limit/ES Sensing** in the **Options** menu, and selecting **No**.
12. For all MSIDs, click in the **Filter Test** column, click **Set Filter Test** from the **Options** menu, and select **None**.

13. Click . On the **Validate** dialog box, click . If you followed the steps accurately, your list should validate without errors. The successful validation message will be displayed in the main window's message area. Close the **Validate** dialog box by clicking

.

14. Click .

After the request is processed, a file called **comptest.sto** is created that can be used by Computation Operation, which is what you will do in the final module of the course.

Review Questions

Indicate the answers for each question below. The correct answers are given immediately following the questions.

These questions cover some theory behind creating a NRT computation.

1. You must enter the MSIDs into the NRT request form in the exact order that they appear in the Input MSID list of your computation. Why?
2. In reference to Question 1, what would happen on a running display that was monitoring a NRT computation if you had requested the MSIDs in a different order than they appear on the Output MSID list in the computation? Short of re-requesting the NRT file, what is the simplest way to rectify this problem?

Review Answers

1. The order that you specify them on the input MSID list is the order that they will be read in the file.
2. You will receive an incorrect output. Change the order of the input MSID list, rebuild the computation, and restart the computation.

Module 8

Running a NRT Computation

Running your NRT computation is as simple as running a real-time computation. In fact, there is only one difference - selecting the input/output type when starting the computation.

This short module guides you through starting your computation using the NRT file you requested. First, however, let's view the file using the Applix spreadsheet software, a commercial-off-the-shelf (COTS) product integrated into the HOSC's software suite. This will allow you to see how data is represented in the file you requested.

Look at Your NRT File

Go to the **Utilities** menu on the Launchpad and click on **Spreadsheet**. A blank Applix spreadsheet will be invoked. Go to the **File** menu and click on **Import** to invoke the Applix **Import** dialog box.

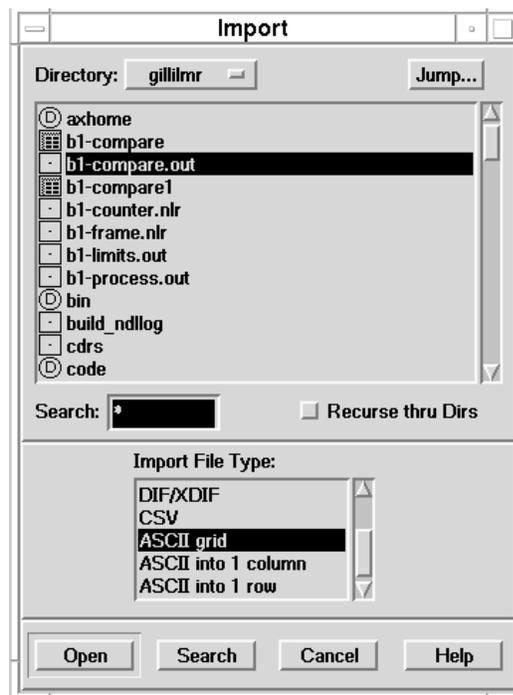
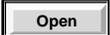


Figure 8-1, Import Dialog box

From the dialog box, select the **sto** directory. This is the EHS subdirectory which contains “standard output” ASCII text files. Select the file matching the output filename you provided in your NRT request (which was **comptest.sto**). The system will automatically identify an appropriate file type. Click on . The request output file will open within the Applix spreadsheet.

What Does it All Mean?

The various pieces of information contained in the file are designated in columns. These columns include Time, MSID, and Status Character.

This information is given as a reference only. Your NRT file may appear significantly different than the one presented here.

The **Time** column shows all times for which you requested data. In the figure below, the request covers a 10-second time slices. The time reference as defined in the request is GMT. Your request should have covered a 4-minute time slice.

The **MSID** column identifies each MSID you included and its relevant data. For the sake of simplicity, The illustrated request included only one MSID (A71P2011A). Your request will include a number of MSIDs.

The **Status Character** column displays a code that represents the status of the data received. No status characters are shown in this request. However, your request may contain status characters.

The following figure illustrates the request referred to in the above paragraphs:

/tmp_mnt/home/team2/team2/gillilmr/b1-compare2.as

File Edit View Style Find Ranges Charts Objects Utilities Window Help

B I U

	A	B	C	D	E	F
1	#Header	HOSC	Receipt	GMT	A71P2011A	status
2	#Data_Type		float		float	
3						
4	#Start_Data					
5	#Data	96	038:15:38:23		1	
6	#Data	96	038:15:38:24			
7	#Data	96	038:15:38:25		1	
8	#Data	96	038:15:38:26			
9	#Data	96	038:15:38:27			
10	#Data	96	038:15:38:28		1	
11	#Data	96	038:15:38:29			
12	#Data	96	038:15:38:30		1	
13	#Data	96	038:15:38:31		1	
14	#Data	96	038:15:38:32		1	
15	#Data	96	038:15:38:33		1	
16	#End_Data					
17						
18						

Figure 8-2, Example NRT Output in Applix

Starting Your NRT Computation

Exercises

Instructions

The following “Try It” allows you to develop a NRT List Request. Carefully read and complete each step.

Try It...

1. Click the **Operation** menu on the **Launchpad** and select **Computation Operation**.
2. Click .
3. In the **Computations** frame of the **Start** dialog box, select the computation you created in Module 5. If you followed the recommended filename structure, you should see a file called “mod5_xxx” where xxx are your initials.
4. In the **Input/Output Type** frame, select the **File** radio button. This causes the **Input File:** selection list to become active. You should see the file **comptest.sto** in this list. Select this file.
5. In the **Database Version** frame, select the latest revision of the TDB.
6. In the **Priority** frame, select the **Medium** radio button.
7. Click .

If all goes as planned, your computation should begin executing and you will see text output (including the value of the pseudo-MSIDs you are writing to) in the Standard Output area of the Computation Operation main window. If you wish to view the results more clearly, re-start the display you created in Module 6.

Review Questions

Indicate the answers for each question below. The correct answers are given immediately following the questions.

These questions cover some theory behind NRT computations.

1. When you selected an input file, you no longer needed to select a data mode. Why?
2. What seems to happen when the computation runs out of data from the NRT file (i.e., it reaches the end of the NRT file)?

Review Answers

1. Your file is now your source of data.
2. The computation terminates.

Course Summary

A computation is a C or FORTRAN executable created by users that can receive telemetry input data (either raw and/or pseudo data), perform operations on the input data, and then output the processed data (called pseudo data). This pseudo data can then be used by other applications internally on the same workstation or externally on different workstations.

In this course, you learned how to construct the components that make up a computation, build the computation, and execute it.

Appendix A

Computation Generation/Operation Questions and Answers

Use the following questions to review the areas covered in this workbook. The answers are also provided in this section.

Computation Generation Questions

1. Name and describe the two types of computations.
2. What four types of data can be used as input for a computation?
3. Define the three types of data processing that can be used when reading in data to a comp?
4. True or False: MSIDs and variable names are not case-sensitive in computations?
5. What menu in the Source Editor provides items which let you quickly move about from one section of your code to another?

Computation Operation Questions

1. What button on Computation Operation's main window do you click to initiate a computation?
2. What type of NRT file can be used to run a computation when data is not available?
3. What are the three data modes? How many channels of playback data are available?
4. Name two ways to stop a computation that is executing.
5. What information is displayed about a running computation on the Computation Operation main window?

Computation Generation Answers

1. User and Mission.

User Computation - creates pseudo MSIDs used on the same workstation or sent out to other workstations on the same project network. The use controls the activation and deactivation.

Mission Computation - creates pseudo MSIDs usually on the project server which are multicasted across the project network. Only privileged accounts can activate or deactivate.

2. Raw downlinked, external pseudo (\$E), internal pseudo (\$I), input computation constants (\$C).

3. Unprocessed - displays raw data

Converted - converts the data to a defined recognizable format

Calibrated - interpolates numbers as number types, point pairs, polynomials, state codes

4. False
5. Navigate

Computation Operation Answers

1. Start
2. NRT List File
3. Realtime, Dump, Playback. 13 playback channels.
4. Push the Stop button on the Computation Operation main window or
Use the process manager to stop it by its associated Process ID (PID).
5. Status, Start-time, UDE, Database, Data mode, Job number.

Abbreviations and Acronym List

A

A/G	Air-to-Ground
A/M	Automatic/Manual
ADQ	Average Data Quality
AIS	Automated Information Security
ANSI	American National Standards Institute
AOS	Acquisition of Signal
API	Application Programming Interface
APID	Application Process Identifier
	Application Process Interface Definition
APT	Active Process Table
AR	Action Request
ASCII	American Standard Code for Information Interchange
AST	Active Server Table
ATT	Attitude

B

BFS	Backup Flight System
BG	Bit-contiguous Group
BPDU	Bitstream Protocol Data Unit

C

C	Counter-dependent
C	C Programming Language
CADU	Channel Access Data Unit
CAP	Command Acceptance Pattern
CAR	Command Acceptance Response
CCBD	Configuration Control Board Directive
CCP	Central Command Processor
CCSDS	Consultative Committee for Space Data Systems
CDB	Command Database
CDD	Command Data Definition
CDQ	Current Data Quality
CLI	Command Line Interface
CM	Configuration Management
CMATS	Configuration Management Asset Tracking System
CMD	Command
CNT	Countdown Time
COTS	Commercial-Off-The-Shelf
CPU	Central Processing Unit
CRC	Circular Redundancy Check
CRR	Command Reaction Response
CSCI	Computer Software Configuration Item
CSM	Command System Management
CSS	Command System Services
CSS	Coarse Sun Sensor

CUI	Common User Interface
D	
DADS	Data Acquisition and Distribution Services
DAE	Data Acquisition and Extraction
DARL	Database Access Routine Library
DB	Database
DBA	Database Administrator
DBCG	Database Coordination Group
DBCR	Database Change Request
DBD	Database Developer
DCM	Document Configuration Management
DCR	Database Change Request
DCRG	Distributed Control Room Graphics
DD AP	Data Distribution Address Processor
DD NS	Data Distribution Network Server
DDQ	Data Data Quality
DDS	Data Distribution System
DEMOS	Distributed Earth Model Orbiter Simulation
DG	Display Generation
DMC	Database Monitor and Control
DMC	Data Management Checklist
DO	Display Operation
DOSH	Database Operational Support History

DP	Distribute Packet
DPU	Data Processing Unit
DQ	Data Quality
DSID	Data Stream Identifier
DSN	Deep Space Network

E

EC	Experiment Computer
ECR	Engineering Change Request
EGSE	Experiment Ground Support Equipment
EHS	Enhanced HOSC System
ELF	Extremely Low Frequency
EM	Exception Monitor
EML	Extract MSID Library
ES	Expected State

F

FDDI	Fiber Distributed Data Interface
FEP	Front-End Processor
FEPSC	Front-End Processor Status and Control
FIFO	First-In-First-Out
FPTNM	Foot-Pounds to Newton-Meters
FSS	Fine Sun Sensor
FSV	Flight System Verifier

FTAM	File Transfer Access and Management
FTP	File Transfer Protocol
G	
GB	Gigabyte
GCID	Ground Correlation Identification
GMT	Greenwich Mean Time
GPC	General Purpose Computer
GPS	Global Positioning System
GSE	Ground Support Equipment
GSFC	Goddard Space Flight Center
GUI	Graphical User Interface
H	
H/W	Hardware
HAMASE	HOSC Automated Model and Screen Editor
HAPS	HOSC Advance Planning System
HASA	HOSC Administrative Software Account
HASS	HOSC Activity Scheduling System
HCR	HOSC Change Request
HLOG	HOSC Automated Logging System
HOSC	Huntsville Operations Support Center
HPR	HOSC Problem Report
HSR	HOSC Support Request

HTT	HOSC Training Team
HUA	HOSC User Assistance
HViDS	HOSC Video Distribution System
HVoDS	HOSC Voice Distribution System
I	
I/O	Input/Output
ICD	Interface Control Document
ID	Identification
IDD	Interface Description Document
IDQ	Instantaneous Data Quality
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
ISS	International Space Station
IST	Integrated Support Team
J	
JSC	Johnson Space Center
K	
kbps	kilobits per second
KMRTS	Kennedy Marshall Redundant Transmission System
KSC	Kennedy Space Center

L

LAN	Local Area Network
LDP	Logical Destination Processor
LES	Limit/Expected State Sensing
LOS	Loss of Signal
LOV	List of Values
LPS	Launch Processing System
LTG	Local Table Generation
LTO	Local Table Operation
M	
M	Multi-syllable
MB	Megabyte
MCC	Mission Control Center
MCCU	Mission Control Center Upgrade
MDM	Multiplexer/Demultiplexer
MET	Mission Elapsed Time
MF	Maintenance Fixtures
MH	Message Handler
MOC	Mission Operations Computer
MOL	Mission Operations Laboratory
MOP	Mission, Operational Support Mode, and Project
MSFC	Marshall Space Flight Center
MSID	Measurement/Stimulus Identifier

MSL	Microgravity Science Laboratory
MTBF	Mean Time Between Failure
MUPS	Momentum Unloading Propulsion System
N	
N	Normal
NASA	National Aeronautics and Space Administration
NASCOM	NASA Communications
NASDA	National Space Development Agency (Japan)
NCC	Network Control Center
NDE	Non-operational Development Environment
NDL	NRT Data Log
NDL	Near Real-Time Data Logger
NEMS	NASA Equipment Management System
NG	Bit Non-contiguous Group
NGT	NASA Ground Terminal
NRT	Near Real-Time
NSOC	NASDA Space Operations Center
NSTS	National Space Transportation System
NTP	Network Time Protocol

O

OCC Operations Control Center

OCDB Operational Command Database

OCR	Operations Change Request
OD	Operational Downlink
OI	Operational Instrumentation
OS	Operating System
OSF	Open Software Foundation
OTE	Operational Test Equipment
P	
PAP	Payload Activity Plan
PB	Playback
PC	Polynomial Coefficient
PC	Personal Computer
PCAD	Pointing Control and Aspect Determination
PCDB	Project Command Database
PCM	PIMS Configuration Management
PDI	Payload Data Interleaver
PDRF	Playback Data Request Form
PDSS	Payload Data System Services
PFS	Primary Flight System
PI	Principal Investigator
PID	Process Identifier
PIMS	Payload Information Management System
POCC	Payload Operations Control Center
POD	Payload Operations Director

POIC	Payload Operations Integration Center
PP	Point Pair
PPS	Payload Planning System
PRT	Packet Routing Table
psi	pounds per square inch
P/SS	PDSS System Support
PTC	Payload Training Complex
PTDB	Project Telemetry Database

R

R	Range-dependent
RAM	Random Access Memory
RCS	Reaction Control System
RDBMS	Relational Database Management System
RDRP	Raw Data Record Playback
RID	Review Item Discrepancy
RPM	Rounds per Minute
RR	Replanning Request
RSS	Resident Size
RT	Real-time
RTAS	Radians to Arcsecs
RTD	Radians to Degrees
RTDS	Real-time Data System
RTS	Requirements Tracking System

RUM	Remote User Machine
RW	Reaction Wheel
S	
S	Super
S&E	Science and Engineering
SC	State Code
SC	Subsystem Computer
SCM	Status and Configuration Manager
SCR	Strip Chart Recorder
SDT	Shuttle Data Tape
SGI	Silicon Graphics Indy™
SGI	Silicon Graphics Incorporated
SL	Spacelab
SM	System Monitor
SMAC	System Monitor and Control
SMCM	System Monitor and Control Configuration Manager
SN	Space Network
SNMP	Simple Network Management Protocol
SOA	Science Operations Area
SPL	Scratchpad Line
SQL	Structured Query Language
SRD	Serial Receive Device
SRS	Software Requirements Specification

SS	System Services
SSCC	Space Station Control Center
SSME	Space Shuttle Main Engine
SSUP	System Services User Profile
STS	Space Transportation System
T	
T	Typical
TAS	Test and Simulation
TBD	To Be Determined
TBS	To Be Supplied
TCP/IP	Transmission Control Protocol/Internet Protocol
TDB	Telemetry Database
TDM	Time Division Multiplexer
TDRSS	Tracking and Data Relay Satellite System
TDS	Time Distribution System
TNS	Telemetry and Network Services
TNSDP	Telemetry and Network Services Distribute Packet
TTY	Teletype
U	
UDE	User-generated Data Element
UDSM	User Data Summary Message
UFT	Unrestricted File Transfer

UGSE	User Ground Support Equipment
UI	User Interface
UPAR	User Profile Access Routine
UPD	User Performance Data
V	
VC	Virtual Channel
VCDU	Virtual Channel Data Unit
VCID	Virtual Channel Identifier
VMS	Virtual Memory System
VV	Verification and Validation
W	
WCP	Workstation Command Processor
WEX	Workstation Executive
WSGT	White Sands Ground Terminal

Glossary

Accelerator	A sequence of keys that provides immediate access to application functions. For example, Ctrl + N to invoke the N ew menu item.
Activation Type	Method used within local table application to activate a group. The defined methods are Time and Control.
Active Window	The workstation window that has input focus and in which keyboard entries impact and may appear. See “Input Focus.”
Analog	A mechanism in which data is represented by continuously variable physical quantities.
Application Main Window	The primary window of a software application.
Application Process Identifier (APID)	The APID is an 11-bit field that is included in Consultative Committee for Space Data Systems (CCSDS) headers. It uniquely identifies the vehicle that created the source packet.
Application Title Bar	The bar at the top of a main window that consists of the window menu button, the title area, and the minimize and maximize buttons.
Apply Pushbutton	A pushbutton that implements any changes made within its dialog box, but leaves the dialog box on the screen so that additional changes can be made. See “OK Pushbutton.”
Approve	In PIMS, this action is taken by a reviewer to signify his/her approval that a document, change request, or data request be placed in the baselined state.
Archived Database	A telemetry database that no longer reflects the current real-time telemetry characteristic information. Only one archive database is available online at a time.
Attributes Defaults Bar	The area below the menu bar on the main window where application Text: , Line: , and Fill: default attributes are set.
Avtec™	A manufacturer of telemetry transmit and receive devices used in the Huntsville Operations Support Center (HOSC) Enhanced HOSC System (EHS) as the primary telemetry processing hardware devices.
Baselined	In PIMS, the final state of the review cycle. When a document, change request, or data request has been approved by all reviewers, the approver may place it in the baselined state.

Baselined Database	Database that reflects the current real-time telemetry or command characteristic information for a particular mission activity. Baselined databases have completed validation.
Bitstream Protocol Data Unit (BPDU)	A protocol data unit of the bitstream function having a format of a header followed by a fixed length block of contiguous bitstream data.
Block	NASA Communications (NASCOM) 4800-bit block format utilized for the transfer of data via the GSFC/MSFC Multiplexer/Demultiplexer (MDM) system.
Calibrated	Three types of calibration exist for telemetry samples: polynomial, point pair interpolation, and state code conversion. If calibration is requested, Telemetry and Network Services (TNS) automatically converts the unprocessed sample and then performs calibration on the sample for that Measurement/Stimulus Identifier (MSID) as defined in the local table.
Cancel Pushbutton	A pushbutton that allows a user to exit a dialog box without implementing any changes.
Cascade Menu	A sub-menu or menu-within-a-menu that appears when you highlight a menu function that has an arrow to the right of its name. Cascade menus are used to group similar functions together beneath the pulldown menu.
Caution	A standard icon used throughout the user guide set to represent destructive actions which could result in loss of data.
Caution Limits	A range defined by a high and low value for an analog MSID in the Telemetry Database (TDB) and Local Table. A color code (yellow) represents values within those ranges in the application.
CCSDS Packet	A source packet comprised of a 6-octet, CCSDS defined primary header followed by an optional secondary header and source data which together may not exceed 65535 octets.
Channel Access Data Unit (CADU)	Protocol data unit used for transmission from the ISS to the PDSS. A CADU consists of a CVCDU that has been prefixed and delimited by a synchronization marker.
Click	The action of pressing and releasing a mouse button. Typically, this is a left mouse button action.
Coded Virtual Channel Data Unit (CVCDU)	A VCDU to which a block of error-correcting Reed Solomon (RS) check symbols has been attached.
Command System Manager	The position in charge of controlling the commanding system utilizing the Command System Management software. For AXAF projects, this is known as the PAYCOM position.

Commercial-Off-The-Shelf (COTS) Software	Software applications that have been purchased from a commercial software vendor as opposed to those that were developed internally.
Computation	A FORTRAN or C program used to manipulate telemetry parameters. These programs are created by the Computation Generation application and are executed in Computation Operation.
Configuration Management (CM) Tools	Institutional applications that allow users to access and perform tasks, such as tracking requirements and equipment, scheduling resources, and logging into automated problem report systems.
Control Indicator	Used to indicate that the group will be activated for limit/expected state (LES) sensing with either the control MSID or a control MSID plus delay time.
Control Panel	The area of a window where application pushbuttons and other graphical components are located.
Converted	The process of translating raw telemetry data into an American National Standards Institute (ANSI) standard data representation so that the sample can be properly interpreted by the machine which processes the data.
Counter-dependent	A parameter whose occurrence in telemetry is dependent on an incrementing or decrementing counter in the data.
Critical Command	A command whose initiation and execution could possibly cause damage to a payload or spacecraft and impair the mission.
Database Administrator	An individual who is primarily responsible for managing the RDBMS engine and administering database accounts. He/she also has the privilege to edit restricted database fields in any database, but is normally not recommended to edit data values that drive the telemetry and command processing for the EHS system.
Database Coordination Group	A working group which includes representatives from the appropriate project operations personnel, project source DB developers, MOL DB developers and the HOSC validation team. Review and approve/dissapprove DBCRs, resolve conflicts and evaluate any DB related issues.
Database Developer	An individual that has the privilege to edit restricted fields (e.g., decom, etc.) for both operational and non-operational databases that drive telemetry and command processing for the EHS system.
Dataset	A saved set of a command's modifiable fields used to update a command prior to being transmitted.

Data Stream Identifier (DSID)	A field within a Secondary EHS Protocol Header for PDSS Payload Data used as a unique identifier for the data stream. This bit denotes if the type of data contained therein is CCSDS packet data (0) or BPDU (1).
Delivered Database	A database must be delivered before it can become pre-released. A delivered database has not been validated for operational testing.
Delta Limit	Maximum acceptable difference between consecutive samples of a parameter.
Desktop	The computer monitor backdrop area on which all windows are opened. May also be referred to as workspace.
Development	In PIMS, the first state of the review cycle in which a document, change request or data request is still being written or is being updated.
Direction Keys	A group of computer keyboard arrow keys which allow users to move up, down, left, and right within an application or menus.
Disapprove	In PIMS, the action taken by a reviewer to signify disapproval and recommendation against moving a document, change request or data request into the baselined state.
Discrete Values	Telemetry values that have states (e.g., on or off).
Double-click	The action of pressing and releasing a mouse button twice in rapid succession.
Drag	To press and hold down a mouse button while moving the mouse on the desktop (and the pointer on the screen). Typically, dragging is used while moving and resizing windows.
Drawing Tools Palette	A group of tool buttons that is used to create graphic objects in order to display telemetry data, initiate commands, and start scripts and computations. The palette is located on its own floating dialog box or the application window.
Dump	During periods when communications with the spacecraft are unavailable, data is recorded onboard and played back during the next period when communications resume. This data, as it is being recorded onboard, is encoded with an onboard embedded time and is referred to as dump data. When a near real-time (NRT) request is written specifying that dump data is desired, the onboard embedded time is used to fulfill the request.
Dynamic Objects	Graphical objects that represent updating telemetry data.
Ellipse	A geometric shape which can be created on a display (i.e., a plane of a cone, an oval shape, etc.).

Expected State	Text state code which indicates the nominal value of a parameter.
Expert Mouse Actions	Clicks or double-clicks of mouse buttons which are non-standard and which activate special functions.
Filter	The filter function is used within a dialog box to refine and define subsets of files you want to work with using a string search and wildcard. Characters can be used to implement the filter function.
Fonts	A style of printed text characters.
Graphical User Interface (GUI)	A way of interacting with computers using graphics-oriented software and hardware.
Grayed out	A menu selection item that has been made insensitive, which is visually shown by making the menu text gray rather than black. Items that are grayed out are not currently available.
Greenwich Mean Time (GMT)	The solar time for the meridian passing through Greenwich, England. It is used as a basis for calculating time throughout most of the world. Displayed within the HOSC, it follows the format ddd:hh:mm:ss.
Grid	A pattern of horizontal and vertical lines forming squares of uniform size on a display, used as a reference for locating points.
Group Parameter Composition	Parameter composition where the bits of a parameter are contiguous and a multiple occurrence of that parameter exists as a group of samples.
Groups	MSIDs which have been grouped together, primarily for use with the Exception Monitor (EM) application.
Hazardous Command	A command whose initiation and execution could pose a threat to human life or the entire mission.
Help	A standard icon used throughout the user guide set to indicate that a cross-reference is provided to assist in solving problems or to answer questions.
Huntsville Operations Support Center (HOSC)	A facility located at the Marshall Space Flight Center (MSFC) that provides scientists and engineers the tools necessary for monitoring, commanding, and controlling various elements of space vehicle, payload, and science experiments. Support consists of real-time operations planning and analysis, inter- and intra-center ground operations coordination, facility and data system resource planning and scheduling, data systems monitor and control operations, and data flow coordination.
I-beam Insertion Bar	A graphical image used to represent the insertion point of text in a text entry area which provides a visual cue that text entry is anticipated by the system.

Icon	A graphical representation of an object on the desktop. Objects can be minimized (iconified) to clear a cluttered workspace, and restored (opened), as needed.
Input Focus	A window or window element that is activated and available for subsequent actions. Input focus is usually indicated by highlighting or changing the color of the activated element.
Input Slider	An input object that allows users to change values of pseudo parameters and computational constants assigned to objects. Pseudos can be used in other applications (i.e., scripts, computations, etc.).
Insensitive	An object or area of an application window that does not have input focus.
Integrated Support Team (IST)	Institutional groups at the HOSC responsible for configuring, monitoring, and resolving problems with computer systems and application software.
Launchpad	A floating menu bar that is used to initiate all HOSC software applications.
Legend	A table that labels parameters plotted on a chart or grid.
Limit Delta	Maximum acceptable difference between consecutive samples of a parameter.
Limit/Expected State Sensing (LES)	A configurable option in Display Operation that allows the user to select whether he wants to see limit violation status or not. The incoming data is compared against the Local Table limits.
Limits	Defined ranges for a measurement which are used to indicate off-nominal conditions: Caution High, Caution Low, Warning High, and Warning Low.
Line Plot	A plot that uses lines to represent the relationships among telemetry values.
Local Table	A subset of the TDB stored on a workstation or server used for telemetry processing.
Maximize Button	A control button that is located to the right of the application title bar. When pressed, this button enlarges the application window to its largest state.
Menu Bar	The area at the top of a window that contains the titles of pull-down menus.
Merge	The combining of data from different sources for a specific time slice. During merge, the best (cleanest) data from each source will be used to create a contiguous segment of data for the specified time slice.
Message Area	The part of the application window where system messages/responses are shown.

Message Dialog Box	An area that provides information, gives the current status of data, asks questions, issues warnings, or draws attention to errors.
Mini-Application	A secondary main window activated from within a main window application.
Minimize button	A control button located to the right of the application title bar. When pressed, it iconifies the window.
Mission, Operational Support Mode, and Project (MOP)	A MOP is what delineates one EHS activity from another. MOP information is available in the common configuration file on every node.
Mnemonic	An underlined character on a menu item, that allows users to initiate the item by typing letters on a keyboard. A user-friendly name used to reference a command residing in the command database.
Mode Independent	Mode Independent is used to describe any process that is not dependent on a data mode.
Modifiable Commands	Commands containing at least one data field which can be updated during operational activities prior to their uplink transmission.
Mouse	A pointing device that is used along with a keyboard in point-and-click user interfaces. The mouse used with HOSC workstations contains three mouse buttons. The left mouse button is used to activate and select items on windows. The middle mouse button is used for move functions. The right mouse button is used to access popup menus.
MSID Text Field	An output object for viewing telemetry containing a label for the telemetry parameter, as well as the current value of the parameter displayed in a specified format (i.e., decimal, hex, octal, binary, American Standard Code for Information Interchange (ASCII), etc.).
Multiple Drawing Mode	A mode that allows users to draw multiple objects of the same type.
Native Data Type	Defined in the database and indicates how the MSID data will be interpreted in the HOSC.
Nominal	A color code indicating expected conditions within defined limits of parameters.
Non-Shareable	A flag has been set to prevent other users from retrieving your User-generated Data Element (UDE) from the UDE Database and using it on their local workstations.
Normal	A telemetered parameter that occurs once per packet.
Note	A standard icon used throughout the user guide set to direct your attention to specific items of concern.

OK Pushbutton	A pushbutton that implements any changes specified within a dialog box. The dialog box is dismissed after this pushbutton has been selected.
Option Menu Button	A pushbutton which, when clicked, displays a menu of related options. The selected option is shown as the pushbutton label.
Output Slider	An object that displays telemetry parameters.
Packet	A data unit comprised of octets that a source application generates.
Parameter Composition	Describes how the bits of a parameter can be arranged in a packet for a sample(s) of that parameter.
Pixmap Object	A picture that can be drawn using the pixmap editor or scanned and assigned using the pixmap editor.
Playback	Playback data can originate either internally or from some other facility. Project servers in the HOSC receive Playback telemetry streams from the HOSC Data Distribution System (DDS) and perform the same processing as would be performed on real-time telemetry streams.
Pointer	Sometimes called the mouse cursor, the pointer shows the location of the mouse on the desktop. The pointer's shape depends on its mode. (e.g., on a window frame, the pointer is an arrowhead; while you are waiting for an action to complete, the pointer becomes clock).
Point Pair Calibration	A measurement which is calibrated using a series of linear segments. The linear segments are defined by a pair of points for each segment. Each point consists of a raw count value and a corresponding engineering unit value.
Pointer Shapes	A graphical shape that a pointer assumes in the drawing mode (e.g., cross-hairs, I-beams, hour-glasses, etc.).
Polynomial Coefficient Calibration	A measurement is calibrated using the following polynomial calibration equation: where: eu - engineering units cnts - counts $eu = COEF0 + (cnts1 \times COEF1) + (cnts2 \times COEF2) + (cnts3 \times COEF3) + (cnts4 \times COEF4) + (cnts5 \times COEF5) + (cnts6 \times COEF6) + (cnts7 \times COEF7) + (cnts8 \times COEF8) + (cnts9 \times COEF9).$
Popup Menu	A menu that is invoked when the right mouse button is clicked. Functions available are the most common and vary from application to application.
Predefined Commands	Commands completely defined prior to an operational activity. Predefined commands contain no modifiable data fields.

Pre-released Database	A database that has been validated for operational testing. It is used to validate UDEs (displays, comps, etc.) prior to the baseline release of the database.
Project Telemetry Database	Contained within the Telemetry Database, includes the telemetry definitions needed to drive HOSC telemetry processing for a specific project/mission. The source of the real-time telemetry processing tables found in the Telemetry Local Table identified by a project/mission/revision prefix. Also included are tables to contain user copy data, an error log, and an Initial Load Table.
Protocol	1: Provides the formulas for passing messages, specifies the details of message formats, and describes how to handle error conditions. More importantly, it allows us to discuss communication standards independent of any particular vendor's network hardware. A communication protocol allows one to specify or understand data communication without depending on detailed knowledge of a particular vendor's network hardware. 2: A term referring to the type of source data used in the construction of an EHS packet. EHS Packet Protocols include: "C" - CCSDS packet, "P" - pseudotelemetry packet, "T" - encapsulated TDM packet, "B" - encapsulated block packet, and "D" - TDS packets.
Pseudo MSID/Parameter	A parameter identification (ID) that has been assigned to contain the output from a computation.
Pseudo Packet	A telemetry packet consisting of external pseudo MSIDs. External pseudo MSIDs are generated (either by EHS computations or scripts), packetized and multicast on the project LAN.
Pulldown Menu	A list or menu of possible options that is hidden under a general phrase and invoked by clicking the left mouse button.
Pushbutton	A control that causes an immediate action. To press a pushbutton on the screen, point to it and click the left mouse button.
Radiobuttons	A group of buttons that allows users to make only one selection at a time. Radiobuttons are small diamond-shaped buttons.
Range-dependent	A parameter whose occurrence in telemetry is dependent on the value of a range parameter.
Real-time Data	Real-time data is telemetered to the HOSC and distributed for immediate use. Real-time telemetry data, received into the HOSC system and written to the NRT log, is indexed by its time stamps and other identifying information. When an NRT request is submitted that covers a particular time slice, this indexing information is used to meet that request.

Recall Text	Area within an application that allows users to input up to 256 characters.
Release	In PIMS, an action taken by the manager of a document or request that releases it from the baselined state back into the development state for modifications.
Resize Borders	The area that surrounds the framed area of an application, and is used to change the height or width of the window.
Resize Handles	Up to eight handles surrounding an object that allows users to resize objects, displays, or windows.
Review	In PIMS, the second state of the review cycle in which the document, change request or data request has been written and submitted for review and approval.
Sample Composition	Describes how the samples of a parameter are arranged in a major frame.
Scatter Plot	A plot that uses unconnected dots to represent the relationships among telemetry values.
Scratchpad Line (SPL) Directives	Provides users with the capability to start and stop displays, computations, and scripts. A user may also uplink and modify commands and update pseudo MSIDs through the use of SPL directives.
Script	A file containing a sequence of directives that can be invoked in a single step.
Scroll Bar	A control that allows the contents of a window area to be displayed without resizing a window or list.
Select Button	The mouse button used for most operations. By default the select button is the left mouse button.
Set Pushbutton	A pushbutton that allows a user to implement changes based on selections made within a dialog box. Reacts like the OK pushbutton and closes the dialog box.
Setup Message Area	The dialog box that allows users to change the number of lines displayed within the message area of the application main window.
Shareable	A flag has been set to allow other users to retrieve your UDE from the UDE Database and use it on their local workstation.
Shotgun	Parameter composition when the bits of a parameter are scattered in multiple non-contiguous words of a major frame.
Slider Box	A graphical component of the scroll bar, which is dragged to provide a different view of the same file, list, or text area.
State Code Calibration	A measurement is converted to a text state code.
Static Object	A graphical object that is not receiving telemetry data.

Status Bar	A feature that allows the viewing of application critical configurations within the main window.
Submit	In PIMS, an action taken by the manager of a document, change request, or data request in which the document or request is placed in a state of review and approval and released from development.
Subset	A collection of measurements from the total measurement set that is bounded as an integer number of octets but does not constitute the packet itself. A mini-packet.
Super	A parameter that occurs more than once per packet.
Superseded Database	If a baselined database already exists for a project and mission, before a new baselined database can be released, the current baselined database is designated as superseded.
Switch MSID	A parameter whose value determines which limit or calibration set will be used for the specified MSID.
TDM	Time Division Multiplexed - a technique for transmitting multiple parameters within a single serial bit stream by interleaving them, one after the other.
Time Indicator	Indicates that a group will be activated based on a defined start time and deactivated based on a defined stop time.
Time Plot	A plot against time containing up to four Y-Axis parameters.
Time Reference	A time format that is represented in either GMT or Mission Elapsed Time (MET).
Time Tag	A time reference marking an event. For example, a parameter goes out-of-limits at 230:16:00:00. The time tag for the out-of-limit event is 230:16:00:00.
Tip	A standard icon used within the HOSC user guide set to indicate that suggestions or hints are provided.
Togglebuttons	Small buttons that can be switched "on" or "off." To switch a togglebutton, point to it and click the select button. Black indicates that the desired attribute is in effect or "on."
Tolerance	Number of times the MSID exceeds the limit value before an EM warning message is issued.
Typical	Parameter composition when the bits of a parameter are contiguous.
Unprocessed	Raw telemetry data.
User-generated Data Element (UDE)	A user-generated file. For example, a display, script, computation, pixmap, etc., is a UDE.

User Data Summary Message (UDSM)	A PDSS Data Quality packet. Information contained in the packet includes: the start/stop time of the UDSM report period, the number of unplanned LOS occurrences, the number of BPDUs or packets per DSID received, the number of VCDU sequence counter errors, and the number of packet sequence counter errors per DSID.
Virtual Channel (VC)	A CCSDS construct whereby a physical communications channel is shared among different users, each of whom uses part of the available bandwidth for a virtual channel of CCSDS CADUs incorporating predetermined CCSDS identifiers for each user.
Virtual Channel Identifier (VCID)	A binary identifier located within the VCDU header, which when concatenated with the spacecraft identifier, uniquely identifies a particular spacecraft virtual channel.
Virtual Channel Data Unit (VCDU)	A CCSDS data set of specific structure and fixed length, which includes CCSDS specified headers and into which user data is packaged for transmission over the space-to-ground link.
Warning Limits	A color code (red) representing limit violations of a parameter.
Wildcard	Placeholders for other characters in a string. Three wildcards are permitted in most HOSC applications. The “*” represents any combination of characters and the “?” represents any single character. A blank can be used to replace a single “*” to indicate “all”. Database applications use Oracle as their basis; therefore, “%” is used like the “*” and an underscore character “_” is used like the “?”. Blank operates the same way in database applications as in other HOSC applications and represents “all.”
Window Menu	The menu that appears when you press the window menu button, which is located to the left of the application title bar on a window frame. Every window has a system menu that enables you to control the position of the window.
Workspace	The area on a terminal where the windows of a user’s environment appear. The workspace is sometimes referred to as a desktop or root window.
XY Plot	A plot that contains one X-Axis and up to four Y- Axes parameters.